

AMENDMENTS TO THE CLAIMS

Claims 1-59 (Cancelled)

Claim 60 (New) A computer-readable storage medium encoded with a compiler apparatus for generating a machine language program for a processor, the processor including a plurality of instruction issue units and a plurality of corresponding execution units, each instruction issue unit issuing instructions to a corresponding execution unit, and each instruction unit including instruction registers for storing the instructions issued to the corresponding execution unit, the compiler apparatus comprising:

a parser unit operable to parse the source program by extracting, from the source program, a reserved word stored in a storage unit and by carrying out a lexical analysis of the source program;

an intermediate code conversion unit operable to receive the parsed source program and convert each statement included in the parsed source program into intermediate codes according to a predetermined rule stored in the storage unit, the intermediate codes including instructions;

an optimization unit operable to receive the intermediate codes and optimize scheduling of the instructions of the intermediate codes by:

scheduling the instructions of the intermediate codes for each instruction cycle of a plurality of instruction cycles without changing dependencies between the instructions of the intermediate codes, each of the instruction cycles being an instruction cycle that executes instructions in parallel using the execution units; and

scheduling the instructions to reduce a hamming distance between two instructions including (i) an instruction in a target instruction cycle, and (ii) an instruction in an instruction cycle that immediately precedes the target instruction cycle, the two instructions being instructions stored in instruction registers of the same instruction unit, the optimization unit being operable to schedule the instructions to reduce the hamming distance of instructions which are scheduled for each of the instruction cycles; and

a code generation unit operable to receive the optimized intermediate codes and convert the optimized intermediate codes into machine language instructions according to a conversion table stored in the storage unit.

Claim 61 (New) The computer-readable recording medium according to claim 60, wherein the optimization unit is operable to optimize the instructions of the intermediate codes by determining an instruction to be executed in the target instruction cycle and determining an instruction issue unit in which the instruction is to be stored so as to reduce a hamming distance between the two instructions when the instructions are scheduled for each of the instruction cycles.

Claim 62 (New) The computer-readable recording medium according to claim 61, wherein the optimization unit is operable to optimize the instruction of the intermediate codes by determining which instruction is to be executed in the target instruction cycle and determining which instruction register of the instruction issue unit storing the instruction is for storing the instruction, to reduce the hamming distance between the two instructions when the instructions are scheduled for each of the instruction cycles.

Claim 63 (New) The computer-readable recording medium according to claim 60, wherein the optimization unit is operable to optimize the instructions of the intermediate codes by scheduling the instructions to reduce a hamming distance between operation codes of the two instructions, the two instructions being stored in instruction registers of the same instruction issue unit.

Claim 64 (New) The computer-readable recording medium according to claim 60, wherein the optimization unit is operable to optimize the instructions of the intermediate codes by scheduling the instructions to reduce a hamming distance between register numbers of the two instructions when the instructions are scheduled for each of the instruction cycles, the two instructions being stored in instruction registers of the same instruction issue unit.

Claims 65 (New) A computer-readable storage medium encoded with a compiler apparatus for generating a machine language program for a processor, the processor including a plurality of instruction issue units and a plurality of corresponding execution

units, and each instruction issue unit issuing instructions to a corresponding execution unit, and each instruction unit including instruction registers for storing the instructions issued to the corresponding execution unit, the compiler apparatus comprising:

a parser unit operable to parse the source program by extracting, from the source program, a reserved word stored in a storage unit and by carrying out a lexical analysis of the source program;

an intermediate code conversion unit operable to receive the parsed source program and convert each statement included in the parsed source program into intermediate codes according to a predetermined rule stored in the storage unit, the intermediate codes including instructions;

an optimization unit operable to receive the intermediate codes and optimize the instructions of the intermediate codes by:

changing, for each instruction cycle of a plurality of instruction cycles, a correspondence between (i) instructions of the intermediate codes to be executed in the same instruction cycle and (ii) the instruction issue units from which the instructions are issued, the optimization unit changing the correspondence without changing dependencies between the instructions of the intermediate codes converted by the intermediate code conversion unit, and each of the instruction cycles being an instruction cycle that executes instructions in parallel using the execution unit; and

changing the correspondence between (i) instructions to be executed in a target instruction cycle and (ii) the instruction issue units from which the instructions are issued, to reduce a hamming distance between two instructions including an instruction to be executed in the target instruction cycle and an instruction in an instruction cycle that immediately precedes the target instruction cycle, the two instructions being instructions stored in instruction registers of the same instruction unit; and

a code generation unit operable to receive the optimized intermediate codes and convert the optimized intermediate codes into machine language instructions according to a conversion table stored in the storage unit.

Claim 66 (New) The computer-readable recording medium according to claim 65, wherein the optimization unit is operable to optimize the instructions of the intermediate

codes by changing the correspondence between (i) the instruction to be executed in the target instruction cycle, and (ii) the instruction issue units from which the instructions are issued, to reduce a sum of hamming distances, each of the hamming distances being calculated between the two instructions, the two instructions being issued to an identical instruction issue unit, and the instructions used to calculate the hamming distances being included in the instruction issue units, respectively.

Claim 67 (New) The computer-readable recording medium according to claim 65, wherein the optimization unit is operable to optimize the instructions of the intermediate codes by changing the correspondence between (i) the instructions to be executed in the target instruction cycle, and (ii) the instruction issue units in which the instructions are issued, to reduce a hamming distance between operation codes of the two instructions.

Claim 68 (New) The computer-readable recording medium according to claim 65, wherein the optimization unit is operable to optimize the instructions of the intermediate codes by changing the correspondence between (i) the instructions to be executed in the target instruction cycle, and (ii) the instruction issue units in which the instructions are issued, to reduce a hamming distance between register numbers of the two instructions.

Claim 69 (New) A method for generating a machine language program for a processor, the processor including a plurality of instruction issue units and a plurality of corresponding execution units, each instruction issue unit issuing instructions to a corresponding execution unit, and each instruction unit including instruction registers for storing the instructions issued to the corresponding execution unit, the method comprising:

- parsing the source program by extracting, from the source program, a reserved word stored in a storage unit and by carrying out a lexical analysis of the source program;

- converting each statement included in the parsed source program into intermediate codes according to a predetermined rule stored in the storage unit, the intermediate codes including instructions;

- optimizing scheduling of the instructions of the intermediate codes by:

scheduling the instructions of the intermediate codes for each instruction cycle of a plurality of instruction cycles without changing dependencies between the instructions of the intermediate codes, each of the instruction cycles being an instruction cycle that executes instructions in parallel using the execution units; and

scheduling the instructions to reduce a hamming distance between two instructions including (i) an instruction in a target instruction cycle, and (ii) an instruction in an instruction cycle that immediately precedes the target instruction cycle, the two instructions being instructions stored in instruction registers of the same instruction unit, and the scheduling of the instructions reducing the hamming distance of instructions scheduled for each of the instruction cycles; and

converting the optimized intermediate codes into machine language instructions according to a conversion table stored in the storage unit.

Claim 70 (New) A method for generating a machine language program for a processor, the processor including a plurality of instruction issue units and a plurality of corresponding execution units, each instruction issue unit issuing instructions to a corresponding execution unit, and each instruction unit including instruction registers for storing the instructions issued to the corresponding execution unit, the method comprising:

parsing the source program by extracting, from the source program, a reserved word stored in a storage unit and by carrying out a lexical analysis of the source program;

converting each statement included in the parsed source program into intermediate codes according to a predetermined rule stored in the storage unit, the intermediate codes including instructions;

optimizing the instructions of the intermediate codes by:

changing, for each instruction cycle of a plurality of instruction cycles, a correspondence between (i) instructions of the intermediate codes to be executed in the same instruction cycle and (ii) the instruction issue units from which the instructions are issued, the optimization unit changing the correspondence without changing dependencies between the instructions of the intermediate codes converted by the

converting of each statement, and each of the instruction cycles being an instruction cycle that executes instructions in parallel using the execution units; and

changing the correspondence between (i) instructions to be executed in a target instruction cycle and (ii) the instruction issue units from which the instructions are issued, to reduce a hamming distance between two instructions including an instruction to be executed in the target instruction cycle and an instruction in an instruction cycle that immediately precedes the target instruction cycle, the two instructions being instructions stored in instruction registers of the same instruction unit; and

converting the optimized intermediate codes into machine language instructions according to a conversion table stored in the storage unit.

Claim 71 (New) A computer-readable storage medium encoded with a compiler program for generating a machine language program for a processor, the processor including a plurality of instruction issue units and a plurality of corresponding execution units, each instruction issue unit issuing instructions to a corresponding execution unit, and each instruction unit including instruction registers for storing the instructions issued to the corresponding execution unit, the compiler program causing a computer to execute a method comprising:

parsing the source program by extracting, from the source program, a reserved word stored in a storage unit and by carrying out a lexical analysis of the source program;

converting each statement included in the parsed source program into intermediate codes according to a predetermined rule stored in the storage unit, the intermediate codes including instructions;

optimizing scheduling of the instructions of the intermediate codes by:

scheduling the instructions of the intermediate codes for each instruction cycle of a plurality of instruction cycles without changing dependencies between the instructions of the intermediate codes, each of the instruction cycles being an instruction cycle that executes instructions in parallel using the execution units; and

scheduling the instructions to reduce a hamming distance between two instructions including (i) an instruction in a target instruction cycle, and (ii) an instruction in an instruction cycle that immediately precedes the target instruction cycle, the two

instructions being instructions stored in instruction registers of the same instruction unit, the scheduling of the instructions reducing the hamming distance of instructions scheduled for each of the instruction cycles; and

converting the optimized intermediate codes into machine language instructions according to a conversion table stored in the storage unit.

Claim 72 (New) A computer-readable storage medium encoded with a compiler program for generating a machine language program for a processor, the processor including a plurality of instruction issue units and a plurality of corresponding execution units, each instruction issue unit issuing instructions to a corresponding execution unit, and each instruction unit including instruction registers for storing the instructions issued to the corresponding execution unit, the compiler program causing a computer to execute a method comprising:

parsing the source program by extracting, from the source program, a reserved word stored in a storage unit and by carrying out a lexical analysis of the source program;

converting each statement included in the parsed source program into intermediate codes according to a predetermined rule stored in the storage unit, the intermediate codes including instructions;

optimizing the instructions of the intermediate codes by:

changing, for each instruction cycle of a plurality of instruction cycles, a correspondence between (i) instructions of the intermediate codes to be executed in the same instruction cycle and (ii) the instruction issue units from which the instructions are issued, the optimization unit changing the correspondence without changing dependencies between the instructions of the intermediate codes converted by the converting of each statement, and each of the instruction cycles being an instruction cycle that executes instructions in parallel using the execution units; and

changing the correspondence between (i) instructions to be executed in a target instruction cycle and (ii) the instruction issue units from which the instructions are issued, to reduce a hamming distance between two instructions including an instruction to be executed in the target instruction cycle and an instruction in an instruction cycle that

immediately precedes the target instruction cycle, the two instructions being instructions stored in instruction registers of the same instruction unit; and

converting the optimized intermediate codes into machine language instructions according to a conversion table stored in the storage unit.